

Practical 95

Lord Implement Conal Selection algorithm:

Theory:

sonal selection algorithm is optimization algorithm inspired by the Cond selection theory is a his-inspired optimization algorithm based on principles of human immune suctem. system.

2) It minics the behavior of B-cells Cantibody which:

- Recognize antigens (Poriegn substances).

- Clone them selves (proliferation)

- Mutale slightly (to improve affinity),

- Survive if they better suggiste. the artingen (nutural selection).

3) Similarly tot convalle trais to avolve better solutions in a computational problem - solving contest.

4) It belongs to the class of Artificial Immune System (AIS) technique which simulate the learning and memory capabilities of the biological

immune system.

s) Conal selection algo is primarily designed to selve optimization pattern of cognition and classification problems

by imitating how 13-colls

at evolve to recognise antigens.

6) Working! i) Initialization: · Fandomly generate a population of candidate solutions called antibodies.
· Each antibody is typically represented as a real-valued to or binary vector. ii) Fitness Evaluation (Affinity (aladation):

Evaluate how good each antibody
is using fitness function.

This measures how well the

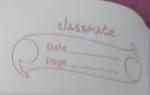
solution matches the problem.

iii) Selection: iii) celection: · Select the top-performing antibodies with highest Pitness scores. (iv) Cloning:
Produce multiple copies (dones) of
solutions. dones an anti-solution might produce. (V) Hypermulation: (som changes) · Apply mutation to each done with a certain probability. · Mutation introduces deversity and explores pearty solutions, allowing the algorithm to escape local optima.

(vi) New population formation:

• Evaluate the fitness of the mutated

clones.



· From the combined set of original solutions and mutated dony select the best individual to form the (vii) Theration: · Repeat stops 2 - 6 for a predefined number of iterations untillering convergence criteria are met The best solution after all iterations is octured as the final optimal solution. 7) trey concepts: · Antigen - problem to be solved · Antibody - solution to the problem. · Affinity / fithness - How good the solution is · Cloning - septicating good solutions to give them more chances. · Hypermutation - Random mutations to dong to discover better solutions. · Selection - treping only the pest solutions.

8) Applications: - Punction optimization,

parameter tuning.

2) Pattern recognition: - Character recognition,

facial recognition.

3) Anomaly detection: - Fraud detection

network intusion detection.



(4) Robotis: - path planning. After we select the top antibodies each selected antibodies is copied.

(None factor times).

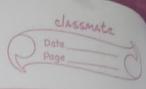
Mutation Browns. 9) Clonning froms: 10) Mutation Proces: -· Mutation means making a small rant random changes to the loves. hanges to each value (gene). Mutation happer with a probability eg: if a done A has gene 0.6 mutation o.65 or 0.55. 1) Selection Process: · Selection means peoping only the best solutions after mutations. · After mutation: we have old population + new mutated dones ((arge number). · We evaluate all individuals based on titness. ones store (based on fitness) to survive for the reset generation. (1) Fitners:-

Solution & is.

The colouted colculated by a fitners function

whigher fitners > better solution.

Fitness is a numerical value that tells us has good a particular



· Fitness function in code: def fitners function (x): yeurn & - np. Sum ((x-0.5)*+2) E 0.2, 0.7, 0.4, 0.8, 0.5) · We call calculate how dose each value in x is to 0.5.
· We square the distance from as
-) (x - 0.5)^2 . Then we sum at these squared · Then we regate the result >
- np. sum ((-.))

· Therefor antibodiy closer to 0.5 get higher fitnes. · Croal is to minimize the distance no 6.5. · Normally, smaller distance means
better solutions.

· But Co. CLONALCS expects higher Fitness = better. distance. (Smaller distance become higher fitness values). In real -world moblem, the target could be anything - 0.5 is just a simple, symmetric point en the range [0-]